Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2016, Article ID 5196092, 12 pages
http://dx.doi.org/10.1155/2016/5196092

Research Article

Bounds on Worst-Case Deadline Failure Probabilities in Controller Area Networks

Michael Short

Electronics & Control Group, Teesside University, Middlesbrough TS1 3BA, UK

Correspondence should be addressed to Michael Short; m.short@tees.ac.uk

Received 6 November 2015; Revised 2 March 2016; Accepted 28 March 2016

Academic Editor: Tin-Yu Wu

Industrial communication networks like the Controller Area Network (CAN) are often required to operate reliably in harsh environments which expose the communication network to random errors. Probabilistic schedulability analysis can employ rich stochastic error models to capture random error behaviors, but this is most often at the expense of increased analysis complexity. In this paper, an efficient method (of time complexity $O(n \log n)$) to bound the message deadline failure probabilities for an industrial CAN network consisting of n periodic/sporadic message transmissions is proposed. The paper develops bounds for Deadline Minus Jitter Monotonic (DMJM) and Earliest Deadline First (EDF) message scheduling techniques. Both random errors and random bursts of errors can be included in the model. Stochastic simulations and a case study considering DMJM and EDF scheduling of an automotive benchmark message set provide validation of the technique and highlight its application.

1. Introduction

Real-time industrial communication networks such as Controller Area Network (CAN) are often required to operate in harsh environments, where they may be subject to environmental hazards such as electromagnetic interference (EMI) and other forms of mechanical/electrical stresses. Exposure to hazards such as this can induce random errors into a system, which, if left uncorrected, may result in system failures [1, 2]. This paper is concerned with probabilistic schedulability analysis of the communications in real-time industrial CAN networks which are scheduled by a priority-driven algorithm in the presence of transient and/or intermittent errors. CAN is a multimaster, differential serial bus using NRZ encoding at the physical layer and is often employed for distributed real-time control applications. The full CAN protocol description may be found in [3]. Although in its native form CAN supports fixed priority scheduling, simple protocols operating at the node level have previously been developed to enable distributed Earliest Deadline First (EDF) scheduling with CAN [4, 5]. As such, the paper focuses upon both fixed priority (the Deadline Minus Jitter Monotonic (DMJM) algorithm) and dynamic priority (EDF) scheduling methods, the latter of which can be enforced at the CAN application layer, due to their known optimality in a networked environment under a wide variety of operating configurations [6, 7]. Specifically, the paper is concerned with the fast calculation of tight bounds on the probability that a deadline will be missed when error arrivals cause messages to be aborted and subsequently rescheduled for transmission after the transmission of an error message.

Such retransmission is a form of redundancy which requires some temporal "slack capacity" in the message schedule; how much slack is required to be allocated depends upon many factors including the level of criticality in the service the system provides, the message set parameters and scheduling algorithm, and also the nature of the error detection and correction mechanisms employed by the system. If insufficient slack is employed by a system to tolerate the effects of the errors it experiences, then aborted messages will not be processed or delivered correctly before their deadlines and system failures may occur. Since error arrivals are random in nature, then no 100% guarantees of timeliness can be given, and probabilistic guarantees are instead sought. In this respect, the principal contribution of the paper is the derivation of an efficient means to obtain tight bounds on the probability of deadline failure when errors occur randomly (with geometrically distributed interarrivals) and

possibly in random bursts (with geometrically distributed burst interarrival and length), and aborted messages are immediately requeued for transmission.

Although previous work related to probabilistic schedulability analysis for wired networks such as CAN and also wireless networks has employed rich stochastic error models to capture these random behaviors, to date this has (mostly) been at the expense of increased analysis complexity. In this paper, some recent results on probabilistic real-time error models and schedulability analysis in [1, 8] are extended, and an efficient method to bound the deadline failure probabilities for a set of n periodic/sporadic messages transmitted in a real-time industrial communication network is proposed. The procedure first carries out an analysis of the available slack in the DMJM or EDF schedule and uses this information, in conjunction with knowledge of the environmental error characteristics, to determine the probability that the slack will be exceeded by the extra load induced by errors. The analysis may be performed in $O(n \log n)$ time. Stochastic simulations and an example related to scheduling a benchmark set of messages on an automotive CAN network are used to illustrate the technique.

The motivation for this bounding method arises principally due to the need for efficient dependability-aware online admission or Quality of Service (QoS) controls in flexible networks (e.g., for automotive applications). In addition, motivation arises from the need for techniques which can provide designers with methods whereby the impacts of different design options (such as the choice of scheduling algorithm and configuration of message parameters and network bandwidth) upon system reliability may be quickly explored at early stages of a design. The remainder of the paper is organized as follows. Section 2 presents a brief summary of related work to contextualize the paper. Sections 3 and 4 present the network and error models, respectively. The proposed technique is outlined in general terms in Section 5 and is applied to DMJM and EDF scheduling in Section 6. Section 7 presents stochastic simulations to validate the proposals, and Section 8 describes a detailed example based upon a benchmark set of messages for an automotive network. Section 9 concludes the paper.

2. Related Work

Although this paper is principally concerned with industrial CAN networks, related work on CPU and communications networks in general is included due to similarities in both the error and task models which are employed. As argued in previous works [1, 8], error models employed in real-time schedulability analysis (for both CPU and wired/wireless network message scheduling) typically assume either (i) a pseudoperiodic arrival of errors or (ii) the fact that some fixed number k of errors will be experienced over a known time interval (e.g., the major cycle of the system operation) or (iii) error arrivals being purely random, typically as a result of a (possibly compound or Markovian) binomial or Poisson process.

Although approach (i) is relatively straightforward to incorporate into an existing schedulability analysis, in many cases it does not effectively capture either randomness (with exceptions, e.g., [9]) or bursty characteristics. Approach (ii), on the other hand, seems well suited to bursty characteristics: the work of [10] presents an exact analysis for CPU task scheduling with EDF in the case where the k errors arrive during the system's major cycle. However the assumption that not more than one burst of k errors will arrive over this known time period implies a deterministic model, which seems inappropriate since errors are due to random noise and interference. In addition, if k errors arrive in some time interval of length proportional to the smallest relative deadline of any task as per [10], then it again seems unjustified to assume that only k errors will arrive in some proportionally much larger time interval (e.g., the major cycle). In [11], a similar method to [10] is employed to analyze the timing properties of wireless channels with bounded retransmissions; however the number of retransmissions can be specified independently for each wireless message and the length of time is made proportional to the message relative deadline. Although this seems a much improved model to use, there is no attempt made to link the retransmission bounds to environmental error models and/or the required reliability of message delivery. Approach (iii) is the one most generally taken for the analysis of distributed systems such as CAN, and the error models employed in these works are typically much richer than those employed for CPU schedulability analysis (although, as noted, the scope of these models is not restricted to the networked environment) [1, 12–15].

Marques et al. analyze the use of servers for the retransmission of messages lost due to errors in time triggered CAN networks [16]. The obtained results indicated that the choice of server and rescheduling policy has a significant influence upon the results, and different choices are appropriate depending upon the metric of interest. The choice of a deadline-based retransmission server was found to minimize the number of lost transmissions (deadline failures). Gujarati and Brandenburg [17] describe a procedure to bound the Failures in Time (FIT) rate for a CAN network experiencing network interference and node failures. In [16, 17], error arrivals were assumed to follow a Poisson distribution.

A recent overview of response-time analysis techniques for CAN is provided in [18], for both the deterministic and probabilistic cases. Although these previous works have examined probabilistic (bursty/sporadic/intermittent) errors and also deterministic errors within combined reliability and schedulability analysis frameworks, as noted in [1, 8, 18], one complication of some of these methods and of the use of probabilistic models in general is the potential complexity of implementation. Typically, an iterative procedure is needed to produce either a probability distribution of response times or a breakdown reliability R beyond which point one or more tasks or messages will miss a deadline. This complexity generally makes the use of probabilistic methods impractical where efficiency is required. Some work to help address this issue was presented in [1, 8], where a rich error model was employed to capture the effects of random errors and bursts of errors and simple closed-form expressions were developed to bound the number of error arrivals in a given interval of

time given knowledge of the environmental conditions and desired reliability of the system.

A second drawback in almost all previous analysis is an implicit assumption that each and every error arrival impacts the schedule by inducing a retransmission of a worst-case frame [18]. This is pessimistic, especially for CAN networks which have early error detection and signaling. The pessimism is compounded for bursts of errors, in which multiple errors arrive with only potentially very small temporal separation; it is almost impossible for each error to induce a worst-case frame length. For bursts of errors, a more likely scenario seems that the first error in the burst sequence induces a frame retransmission, where subsequent errors in the burst sequence delay its starting time until the burst subsides. This is not taken into account in [1] or [8], or in any of the methods described in [18]. In the current work, efforts are made to improve the analysis in an attempt to address this issue, whilst retaining most aspects of the simplicity of application.

3. CAN Network Model

In the analysis that follows, it is taken that time is discrete (one time unit will typically, but not necessarily, correspond to one network bit-time) and is indexed by a nonnegative integer variable t. It is assumed that the system consists of a number of distributed nodes, which share a communications channel to exchange real-time messages. A standard model for a shared real-time communication system is adopted to represent the CAN network in that the system to be implemented can be represented by a set of n messages $\tau_1, \tau_2, \ldots, \tau_n$. Each message is represented by 4-tuple:

$$\tau_i = \left(T_i, C_i, D_i, J_i\right), \tag{1}$$

in which T_i is the message period/interarrival, C_i is the worst-case transmission time of any instance of the message, D_i is the message relative deadline, and J_i is the jitter (the worst-case time that may elapse between an (external or internal) event occurring initiating a message transmission in a distributed node and it being released to the network for transmission). Jitter is typically induced by variation in the latencies of distributed node event handlers. For CAN messages, either standard (11 bits) or extended (29 bits) identifiers can be used; as is well-known (e.g., [19]), for a message with Data Length Code (DLC) $\in [0, 8]$, the total number of bits for a message with an 11-bit identifier is given by $C = 55 + 10$ DLC. For 29-bit identifiers, $C = 80 + 10$ DLC. For CAN networks, the worst-case length of an error frame $C_E = 31$ bits.

The utilization of an individual message is given by $U_i = C_i/T_i$ and represents the fraction of time the network will be occupied processing the frames generated from the message over its lifetime. Successive frame arrivals from sporadic messages are invoked by both internal and external events (typically hardware or software interrupts on their host node) and are always separated by at least T_i units of time; frame arrivals from periodic messages are always separated by exactly T_i time units on their host node and are invoked by a logical timer, which may possibly be synchronized to a distributed clock. It is assumed that worst-case clock synchronization errors between any host nodes can be incorporated into these jitters and the synchronization protocol messages modeled as regular network traffic.

When a frame of message i arrives (becomes ready) at some time t, its *absolute* deadline is set at time $t + D_i$ and the scheduling procedure must allocate C_i units of network time to process the job in the interval $[t, t + D_i)$; otherwise a deadline miss will occur. Both fixed priority (DMJM) and dynamic priority (EDF) scheduling procedures are considered in this paper, covering a wide variety of wired and wireless industrial network protocols and protocol extensions. It must be cautioned at this point that fixed priority and EDF scheduling, aside from a limited number of specific cases, are seldom supported directly at the MAC level. A drawback is that their use inevitably requires overlay protocols leading to the introduction of low-level overheads; however, the many recognized benefits (such as enabling higher network utilization and facilitating easier temporal analysis) oftentimes outweigh this drawback. CAN in its native form supports fixed priority scheduling, and hence DMJM is supported by design. Descriptions of overlay protocols enabling priority-based scheduling in Flexible Time Triggered (FTT) master-slave systems for both switched Ethernet and wireless networks can be found in [20, 21]. Extensions to enable EDF scheduling in wireless and wired networks using IEEE 802.15.4, Bluetooth, and CAN are discussed in [4, 5, 11, 22], respectively. Prospective system designers considering the use of any overlay protocol should be sure that the benefits outweigh the overheads for their particular application. Consideration of such analysis is outside the scope of the current paper.

Although the techniques described in this paper can be applied (in principle) to a message set with arbitrary deadlines, the main focus is on constrained deadline messages, in which $(D_i - J_i) \leq T_i$. According to previous work, it is known that the worst-case arrival pattern of the message frames in terms of network load is the one in which all frames experience their worst-case jitter simultaneously, such that they are aligned in a synchronous release pattern, and thereafter the frames arrive at their maximum allowed rates [7, 23]. As most real-time networks inherently support nonpreemptive frame transmissions, worst-case blocking due to priority inversion is also required to be accounted for in the timing and schedulability analysis. Henceforth, for convenience it is assumed that the messages are sorted in order of nondecreasing relative deadline minus jitter, that is, for any two messages τ_i and τ_j, if $i < j$ then $(D_i - J_i) \leq (D_j - J_j)$, and that the total network utilization is not greater than unity. It is also assumed that, under EDF scheduling, deadline ties are broken by lowest message index.

4. Error Model

To incorporate the effects of frame errors and retransmissions into a timing analysis, it is necessary to first state some assumptions regarding the effects of errors. As mentioned in the previous section, in almost all previous works it has

been assumed that all errors manifest themselves in such a way that the last bit of the longest valid message frame from the (sub)set of messages currently under analysis is repeatedly corrupted by errors, forcing retransmission of the entire message. In this paper, this basic assumption is relaxed since the basic operation of the CAN protocol is intended to support early detection and signaling of many types of errors and early abandonment of corrupt transmissions, more so than similar serial protocols [19]. At this point it is worth recalling from the CAN protocol definition two important points related to error detection: (i) the probability of an undetected error is vanishingly small (of the order $\gamma \times 4.7 \times 10^{-11}$, where γ is the message error rate) and (ii) in addition to CRC failures, instantaneous bit errors, bit stuffing errors, and form errors may be detected and signaled at any point in a frame transmission by any node (whether transmitter or receiver) [3].

Although in reality the probability of undetected errors is higher than specified in the CAN protocol due to interactions between bit stuffing and the CRC [24] and the fact that inconsistent omissions can occur, let us proceed under the assumption that for detected errors there is a uniform probability for their detection and signaling along the length of a CAN message. The justification for this is as follows. Due to transmitter bit error monitoring, detected errors affecting any set of nodes which includes the transmitting node (including all global errors) are immediately detected and signaled. Local errors affecting a set of nodes which does not include the transmitting node, but only one or more receivers, typically result in form or bit stuffing violations which are immediately signaled as errors by the effected receivers. Detailed simulations and experiments have indicated that well over 99.9% of detected CAN errors will be signaled by these three primary mechanisms [19, 24]. The remaining proportion of detected local errors will principally manifest as CRC failures, signaled only after the CRC check. Thus, although this assumption of uniform probability for detecting and signaling errors along the length of a CAN message is clearly not 100% accurate, it would seem to be close enough for analysis purposes. Let us initially proceed as such.

4.1. Random Errors and Bursts of Errors. As previously mentioned, research has shown that many errors in CAN communication links are not just single independent events but are likely to occur in isolated transient bursts [19, 24, 25]. In order to develop a technique to effectively capture such behaviors, an error model that is rich enough to capture this bursty nature yet simple enough to lend itself to tractable analysis is required. A common way to model bursty behavior is to use a simple two-state discrete Markov model [1, 8, 26, 27], such as is shown in Figure 1.

The model has two states G and B, representing "Good" and "Burst" states, respectively. Transitions between the two states G and B have associated with them static probabilities p_{GB} and p_{BG}. The probability of remaining in a given state is then given by $p_{GG} = 1 - p_{GB}$ and $p_{BB} = 1 - p_{BG}$. Each state may also have associated with it a probability of bit error occurrence (denoted β_G and β_B, resp.). The model parameters p_{GB} and p_{BG} can be interpreted as follows: the reciprocal of p_{GB} defines the expected (mean) gap between error bursts μ_{EG}, and the reciprocal of p_{BG} defines the expected (mean) duration of error bursts μ_{EB}, both variables having a geometric distribution. The expected interarrival time of error bursts is then given as $(\mu_{EG} + \mu_{EB})$. The optional model parameters β_G and β_B can be interpreted as follows: when the system is in the "Good" state, the reciprocal of β_G defines the expected (mean) interarrival time of errors, and when the system is in the "Burst" state, the reciprocal of β_B defines the expected (mean) interarrival time of errors, both variables again having a geometric distribution. Typically, it would be the case that $\beta_G \ll \beta_B$ and $\mu_{EB} \ll \mu_{EG}$. Let the state of the link at step k be denoted by $l(k) \in \{\text{"G"}, \text{"B"}\}$, and let the state of the Markov model at step k be encoded as the probability $s(k)$ that the link is in the error state; that is, $s(k) = P\{l(k) = \text{"B"}\}$. Applying the normal rules for Markov model state transitions, then $s(k)$ depends only upon the previous state $s(k-1)$ and the transition probabilities p_{BB} and p_{GB} and can be recursively computed as follows:

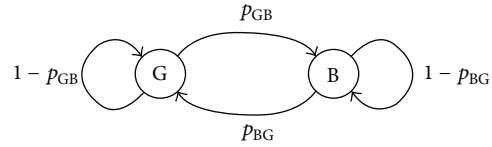

Figure 1: Two-state Markov model for burst errors.

$$s(k) = p_{BB} \cdot s(k-1) + p_{GB} \cdot (1 - s(k-1)). \quad (2)$$

Assuming that the initial state of the link $s(0)$ is known, the transient and steady-state evolution of the link state can be calculated using (2). The steady-state solution to the Markov chain is obtained by first setting $s(k) = s(k-1) = \pi$ into (2). Solving for π, we obtain that $\pi = p_{BG}/(1 - p_{BB} - p_{GB})$. Given any starting state $s(0)$, after a transient period the model converges upon π which represents the fraction of time the link can be expected to be in the "Bad" state when observed for long periods. π also represents the probability that when the link is observed at some random sample time t we find that $l(t) = \text{"B"}$; following the acquisition of explicit knowledge about the link state at step t (such as by making an observation), the state transiently moves back to the steady-state π with coefficient $\alpha = (p_{BB} - p_{GB})$ according to the relationship $s(t+k) = \pi + (s(t) - \pi) \cdot \alpha^k$. The special case in which errors do not arrive in bursts but only have a constant bit error rate of β is covered by setting $p_{BB} = p_{GB} = \beta$, after which we may see that $\alpha = 0$.

For simplicity, in this paper it is assumed that β_B and β_G were set to 1 and 0, respectively, under the assumption that the probability of a full CAN frame transmission occurring whilst the link is in the burst state is effectively negligible (e.g., if λ_B is 0.5, a realistic assumption for a network such as CAN, the probability of successfully sending a 60-bit frame whilst in a burst state is of the order 10^{-20}). In such circumstances π represents the BER of the CAN network. Although other studies have suggested that errors and bursts of errors in CAN networks follow exponential distributions (and can be considered as a Poisson process

which is possibly nonhomogeneous and/or generalized), the model above should be at least as descriptive. This is because (i) if multiple errors arrive during the same logical bit-time, as may be predicted with a Poisson process, then only one of these errors will be effectively "counted" since time is discrete in a CAN network and (ii) the geometric distribution is the discrete equivalent to the continuous exponential distribution. Maximum likelihood estimates of the model parameters p_{GB} and p_{BG} from observed data traces are known and have simple closed forms [26, 27].

4.2. Error Statistics. As will be described in the next section, a means to determine the mean and variance of the additional load on the network which could be introduced by random errors and bursts of errors is required. Let the additional fault load over a time duration of length t be denoted by the random variable $F(t)$, which is the sum of the additional loads introduced due to errors at each individual time step, which are denoted as $e(1), e(2), \ldots, e(k), \ldots, e(t-1), e(t)$. Then we may examine the expected additional load at every time step $e(k)$ and also its variance; the mean and variance of $F(t)$ may then be formed accordingly. From discussions in the previous section, although the BER of the link is π, the impact of an individual error depends to a certain extent upon the arrivals of previous errors. At each time step k, assuming an error arrives, either of the following (mutually exclusive) events could occur:

(1) The link was in the "Good" state at step $k-1$, with probability $(1-\pi)$, but transits to the "Bad" state at step k with probability p_{GB}; the resulting error forces the current transmission to abort and an error frame to commence transmission (denoted as a "type 1" error).

(2) The link was in the "Bad" state at step $k-1$, with probability π, and remains in the "Bad" state at step k with probability p_{BB}; the resulting error delays by one time unit the (on-going) transmission attempt of the error frame following on from the last "type 1 error" described above (denoted as a "type 2" error).

Let us denote the probability of a type 1 error occurring by $p_g = (1-\pi)p_{\mathrm{GB}}$ and the probability of a type two error occurring as $p_b = \pi p_{\mathrm{BB}}$. Note that $p_g + p_b = \pi$; that is, the BER consists of a proportion of each of the two error types. Let the longest frame transmitted during the considered interval of length t be $C > 0$. Since the assumption is that an error during message transmission is immediately detected and the transmission aborted, a type 1 error leads to an increase in the error load by a value uniformly drawn from the interval $[1, C]$, with mean value $(C+1)/2$, plus an error frame of length C_E. A type 2 error, since the link was previously in the error state, simply leads to a unit increase in error load. Denoting expectation of a variable x as $E[x]$, the mean load introduced at each time step can be written in closed form:

$$E\left[e\left(k\right)\right] = p_g \cdot \left(\frac{C+1}{2} + C_E\right) + p_b. \tag{3}$$

An expression for the variance of $e(k)$ can also be written using the "mean of square minus square of the mean" rule:

$$\begin{aligned}\mathrm{Var}\left(e\left(k\right)\right) &= E\left[e\left(k\right)^2\right] - E\left[e\left(k\right)\right]^2 \\ &= \left[\frac{p_g}{C} \cdot \left(\sum_{i=1}^{C}\left(i + C_E\right)^2\right) + p_b \cdot 1^2\right] - E\left[e\left(k\right)\right]^2 \\ &= \left[\frac{p_g}{C} \cdot \left(\sum_{i=1}^{C} i^2 + \sum_{i=1}^{C} C_E^2 + 2 \cdot \sum_{i=1}^{C} i \cdot C_E\right) + p_b\right] \\ &\quad - E\left[e\left(k\right)\right]^2.\end{aligned} \tag{4}$$

To simplify the above expression further, well-known formulae for the sum of the first C natural numbers and their squares can be applied:

$$\begin{aligned}&\left(\sum_{i=1}^{C} i^2 + \sum_{i=1}^{C} C_E^2 + 2 \cdot \sum_{i=1}^{C} i \cdot C_E\right) \\ &\quad = \left(\frac{C^3}{3} + \frac{C^2}{2} + \frac{C}{6} + C \cdot C_E^2 + C_E \cdot C \cdot (C+1)\right).\end{aligned} \tag{5}$$

Giving the following simple closed-form expression for the variance,

$$\begin{aligned}\mathrm{Var}\left(e\left(k\right)\right) = p_g \cdot &\left(\frac{C^2}{3} + \frac{C}{2} + \frac{1}{6} + C_E^2 + C_E \cdot (C+1)\right) \\ &+ p_b - E\left[e\left(k\right)\right]^2.\end{aligned} \tag{6}$$

Using (3) and (6), the mean and variance of the additional load $F(t)$ due to errors can be formed through summation. Observe that there is some weak dependence between the random variables $e(k)$ (in the sense that if the value of $e(k)$ is revealed, the probabilities associated with the values which $e(k+1)$ can assume are influenced). This is present in all probabilistic timing analysis for CAN, due to the serialization of message transmission and the error detection mechanisms employed. The main dependence in the current context is such that a type 1 error cannot immediately be followed by another type 1 error; hence if $e(k) > 1$, $e(k+1) \leq 1$. This manifests itself as an exponentially decaying negative correlation between a variable at step k and its successors at steps $> k$. However, as in previous works on CAN, let us drop this correlation and proceed with the assumption of statistical independence. This is a safe assumption since we now allow the situation in which both $e(k)$ and $e(k+1)$ can be > 1 simultaneously, which cannot occur in practice due to the serialization of CAN message transmission. Although this is pessimistic it facilitates a simpler analysis in the sequel. Briefly returning again to the point related to nonuniform probability for detecting and signaling errors along the length of a CAN message before leaving this section, please refer to Appendix A for details of the modification necessary to (3) and (6) should this require to be modeled.

5. A Simple Deadline Failure Bound

5.1. Probability Inequality. In this paper, the following bound for a sum of independent random variables is to be employed.

Theorem 1. *Let $x(1), x(2), \ldots, x(k), \ldots, x(t-1), x(t)$ be independent random variables. Let each of the $x(k)$ be bounded such that $|x(k) - E[x(k)]| \leq M$, with M some nonzero (positive) constant. Denote the sum of the t variables along with its mean and variance as follows:*

$$X = \sum_{k=1}^{t} x(k),$$

$$\mu = \sum_{k=1}^{t} E[x(k)], \tag{7}$$

$$\sigma^2 = \sum_{k=1}^{t} \mathrm{Var}(x(k)).$$

Then given some real $q \geq 0$, the upper tail of the distribution function of X is bounded by

$$P(X > \mu + q) < \exp\left\{-H\left(\sigma^2, q, M\right)\right\}, \tag{8a}$$

where

$$H\left(\sigma^2, q, M\right) = \frac{\sigma^2}{M^2} \cdot \ln\left(1 + \frac{Mq}{\sigma^2}\right) \cdot \left(1 + \frac{Mq}{\sigma^2}\right) - \frac{q}{M}. \tag{8b}$$

Proof. Consider Equation 8b in Bennett (1962) [28]. □

The expression above is presented in a slightly different form to that found in [28]; however the difference is purely cosmetic. In the current context, t represents the length of an interval under consideration, X represents the additional fault load $F(t)$ which may be observed in this interval, and each of the $x(k)$ represents the additional error loads which could be introduced at each time step $e(k)$. The bound M represents the worst-case length of a message that may be affected by a type 1 error plus the worst-case length of an error frame (C+C_e). With this result recalled, the proposed method may now be outlined.

5.2. Proposed Method. The method takes as input a set of n messages $\tau_1, \tau_2, \ldots, \tau_n$, each with parameters as given by (1), environmental error characteristics p_g and p_b, and the worst-case length of error frame C_e and produces a set of n deadline failure probabilities $p_1^{\text{fail}}, p_2^{\text{fail}}, \ldots, p_n^{\text{fail}}$. The method can be stated in three basic steps which are as follows:

(1) For each message i, carry out a sensitivity analysis to find the critical slack S_i that any generated instance of the message has between its release time and its deadline such that the deadline is not missed, and determine M_i which is the worst-case length of frame that could be transmitted during the busy period plus the length of error frame C_e. If for any message $S_i < 0$, then set $p_i^{\text{fail}} = 1$ and stop as the message is not deterministically schedulable.

(2) For each message i, compute the quantities:

$$t_i = (D_i - J_i), \tag{9a}$$

$$\mu_i = t_i \cdot E[e(k)], \tag{9b}$$

$$\sigma_i^2 = t_i \cdot \mathrm{Var}(e(k)), \tag{9c}$$

with the mean and variance of $e(k)$ computed according to (3) and (6), using $C = (M_i - C_e)$ in both of these expressions. If for any message $\mu_i > S_i$ set $p_{\text{fail}}^i > 0.5$ and exit as an error load which is less than the mean (expected) load which is sufficient to cause a deadline miss.

(3) For each message i calculate $H_i = H(\sigma_i^2, (S_i - \mu_i), M_i)$ using (8b), and bound the deadline failure probability p_i^{fail} using the inequality of (8a).

The rationale for the method outlined above is that, with knowledge of the most critically loaded intervals of the schedule, one may calculate the critical error load that may be tolerated during that interval to (just) maintain schedulability; this is performed using sensitivity analysis (step (1)). Given knowledge of the length of the considered intervals and the environmental error characteristics, the mean and variance of the error load may be determined (step (2)); this allows the application of the inequality given in (8a) in step (3) to determine the probability the error load exceeds the critical error load. The p_i^{fail} values as calculated in this way can easily be linked to higher-level reliability measures in that knowledge of the number of deadlines of each message occurring per unit operation time (e.g., an hour) can be used to calculate the probability that no deadlines will be missed per hour (i.e., the network reliability). In step (1) of the above, both S_i and M_i are required; their determination is the subject of the next section.

6. Application to Fixed and Dynamic Priority Scheduling

As mentioned in Introduction, the paper focuses upon the DMJM and EDF scheduling methods due to their known optimality as fixed priority assignment rule and dynamic priority scheduling algorithm, respectively (albeit the former is for constrained deadlines only), on a single processor or communications network [6, 7]. In order to develop a means to solve step (1) in the proposed method, it is necessary to make some further assumptions. These assumptions are such that all errors are detected and that all messages which fail due to an error are immediately rescheduled using their original priority or deadline. Although the worst-case error load tolerable for a given message can be calculated exactly using sensitivity analysis combined with an exact test like response-time analysis [18, 23] or the processor demand approach [7], both would require an application of an exact schedulability analysis. Since the exact analysis is known to be coNP-complete in both cases [29, 30], simple bounds which may be

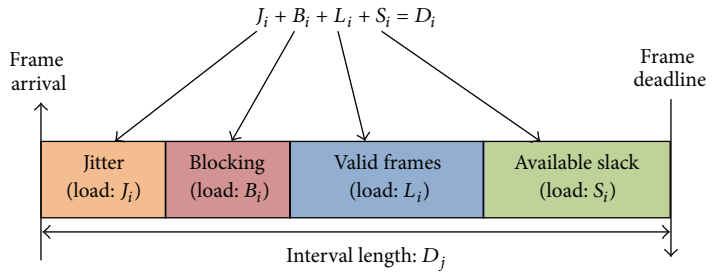

Figure 2: Illustration of the proposed slack analysis technique.

easily manipulated algebraically are instead sought. As such, sufficient and easily computable schedulability conditions will instead be employed.

For both DMJM and EDF, the technique proceeds as illustrated in Figure 2: assuming that a busy period of length D_i occurs, existing sufficient schedulability conditions will be adapted to upper-bound the worst-case workload induced by jitter and priority inversion (blocking) plus the transmission of valid frames which can occur. Assuming that this workload does not exceed the length of the interval D_i (in which case the network cannot be deemed to be deterministically schedulable with the utilized condition), the remaining slack time is assumed to be assigned for error handling and message retransmission purposes.

6.1. Fixed Priority Scheduling. For fixed priority scheduling, the following sufficient condition can be adapted from the relationships obtained by Davis and Burns [23] to verify schedulability under the DMJM priority assignment:

$$\forall i = 1, \ldots, n: \quad J_i + \sum_{j=1}^{i-1} U_j J_j + B_i + C_i + D_i \sum_{j=1}^{i-1} U_j + \sum_{j=1}^{i-1} C_j \left(1 - U_j\right) \leq D_i, \tag{10}$$

where the blocking term $B_i \geq 0$ represents the worst-case length of time a message of priority i may be blocked by a lower priority frame. For nonpreemptive frame transmission under fixed priority scheduling, the blocking term is given by the largest transmission time of any message instance with priority less than or equal to i:

$$B_i^{\mathrm{DMJM}} = \max_{j \geq i} \left\{C_j\right\}. \tag{11}$$

Since interference from messages being retransmitted due to errors appears in (10) in a similar fashion to nonpreemptive blocking (i.e., a lengthening of the busy period), the slack term S_i can be added into the left hand side of (10). Now, setting L_i and J_i as

$$L_i^{\mathrm{DMJM}} = C_i + D_i \sum_{j=1}^{i-1} U_j + \sum_{j=1}^{i-1} C_j \left(1 - U_j\right) \tag{12a}$$

$$J_i^{\mathrm{DMJM}} = J_i + \sum_{j=1}^{i-1} U_j J_j \tag{12b}$$

and assuming an equality in (10) allow one to solve for the slack terms S_i. Given the fixed priority ordering of message transmissions the retransmission bound lengths M_i are also easily obtained, giving

$$\forall i = 1, \ldots, n: \quad S_i = D_i - J_i^{\mathrm{DMJM}} - B_i^{\mathrm{DMJM}} - L_i^{\mathrm{DMJM}}, \tag{13a}$$

$$M_i = \max_{j \leq i} \left\{C_j\right\} + C_E. \tag{13b}$$

6.2. Dynamic Priority Scheduling. For dynamic priority scheduling, the following sufficient condition can be used to verify schedulability under the EDF priority assignment:

$$\forall i = 1, \ldots, n: \quad J_i \left(1 - \sum_{j=1}^{i} U_j\right) + \sum_{j=1}^{i} U_j J_j + B_i + D_i \sum_{j=1}^{i} U_j + \sum_{j=1}^{i} U_j \left(T_j - D_j\right) \leq D_i, \tag{14}$$

where the blocking term $B_i \geq 0$ again represents the worst-case length of time a message of priority i may be blocked by a lower priority frame; for nonpreemptive frame transmission under EDF scheduling with jitter, the blocking term is given by the largest transmission time of any message instance that has a deadline minus jitter greater than the deadline minus jitter of message i [7]:

$$B_i^{\mathrm{EDF}} = \max_{(D_j - J_j) > (D_i - J_i)} \left\{C_j\right\}. \tag{15}$$

The sufficient condition in (14) above was motivated by the work of Devi [31]; however it includes jitter terms: the proof of correctness of (14) is included as Appendix A in the current paper. Now, using (14) allows proceeding along similar lines to the DMJM case. Again, since interference from messages being retransmitted due to errors appears in (14) in a similar fashion to nonpreemptive blocking, the slack term S_i can be added into the left hand side of (10). Again setting L_i and J_i as

$$L_i^{\mathrm{EDF}} = D_i \sum_{j=1}^{i} U_j + \sum_{j=1}^{i} U_j \left(T_j - D_j\right), \tag{16a}$$

$$J_i^{\mathrm{EDF}} = J_i \left(1 - \sum_{j=1}^{i} U_j\right) + \sum_{j=1}^{i} U_j J_j \tag{16b}$$

Table 1: Comparison of analytical and stochastic simulation results.

	Mean burst interarrival	1000	10000	20000	30000
	Mean burst length	1	10	20	30
BER	Calculated	0.001000000	0.001000000	0.001000000	0.001000000
	Observed	0.001000200	0.000999700	0.001000100	0.001003200
Mean	Calculated	0.099000000	0.010800000	0.005900000	0.004266667
	Observed	0.099227000	0.010800600	0.005889200	0.004270500
Var	Calculated	11.309865667	1.132750027	0.566898523	0.378270684
	Observed	11.398210000	1.134113000	0.566252100	0.378072900
p^{fail}	Old bound [8]	0.139636696	>0.5	>0.5	>0.5
	New bound	0.049658914	0.000322315	0.000069962	0.000028680
	Observed	0.000439610	0.000008529	0.000002429	0.000001000

and assuming an equality in (14) allow one to solve for the slack terms S_i. Given the deadline-based priority ordering the retransmission bound lengths M_i are also easily obtained, giving

$$\forall i = 1, \ldots, n: \quad S_i = D_i - J_i^{\text{EDF}} - B_i^{\text{EDF}} - L_i^{\text{EDF}} \tag{17a}$$

$$M_i = \max_{(D_j - J_j) \leq (D_i - J_i)} \left\{ C_j \right\} + C_E. \tag{17b}$$

The methods proposed above allow the completion of step (1) in the proposed method for both forms of scheduling with a time complexity $O(n \log n)$ and space complexity $O(n)$ as per the original schedulability conditions in [23, 31]. Since steps (2) and (3) both require constant time per message to complete, the overall method is bounded in time and space complexity $O(n \log n)$ and $O(n)$, respectively.

7. Stochastic Simulation Studies

In order to evaluate the accuracy and predictive abilities of the probability computations detailed in previous sections, a series of stochastic simulations was carried out to compare calculated values to statistical results. This was deemed important since the analysis in previous work [8] resulted in unacceptable pessimism; hence statistical evaluation can help to validate the improved analysis and assumptions made. The simulations were carried out to assess the measured message deadline failure probability for a single CAN message in the presence of both static bit errors and bursts of bit errors. A message length of 135 bits was assumed (DLC of 8 with 11-bit identifier), with a 31-bit error message length. The message relative deadline was taken to be 500 bits, with the CAN network speed assumed to be 1 Mbps. The stochastic simulator was written in C++.

A total of four simulation experiments were performed. In each experiment, which lasted for 5 simulated hours, the empirical BER, mean, and variance were measured alongside the empirical deadline failure probability for a given set of error characteristics. In each case, the theoretical BER was set to 0.001 and the mean burst length set to either 1, 10, 20, or 30 bits. The first case corresponds to a static BER, whilst the latter three correspond to bursty environments. The chosen BER is high, corresponding to a very noisy environment; this was done to allow meaningful statistical results to be observed without incurring inordinately large simulation times. For each experiment configuration, the mean and variance in the error load per bit were calculated using (3) and (6), with the deadline failure probability obtained using (8a) and (8b). The slack S was taken to be 365 bits. The empirical and analytical results are as shown in Table 1. For comparative purposes, the deadline failure probability obtained by application of the methods reported in [8] is also displayed.

Observing the data reported in the table, one immediately sees a very close correspondence between the measured (empirical) values of BER, mean load, and its variance as compared to the calculated values. This gives some validation in the assumptions made in Section 5 and gives confidence in the use of (3) and (6) to capture the effects of errors in the CAN network. In addition, one sees that although the empirical BER in each case is virtually identical to its theoretical value of 0.001, the mean load and its variance are both reducing with an increase in the length of bursts. This captures the notion that not all errors have the same impact upon the schedule and that "type 1" errors are worse that "type 2" errors from a timing perspective.

Focusing attention now upon the use of (8a) and (8b) to bound the deadline failure rate, it can be observed that although some inaccuracy is necessarily present, it seems effective at capturing the relative order of magnitude of the empirically observed deadline failure probability. In particular, it captures the behavior that, for a constant BER, the deadline failure probability is reducing for increasing mean burst length and this reduction can be quite pronounced. In addition, the pessimism in the bound reduces as the failure probability decreases (as is typical for exponential inequalities). In contrast the previous method described in [8] fails to capture this behavior and the level of pessimism is significantly increased; in fact the method fails to give meaningful results when bursty behavior is considered and asserts only that the failure probability is > 0.5. This is due to the assumption that each error impacts upon the fault load in an identical way, such that only two errors may be tolerated before deadline failure occurs. When the mean burst length is either 10, 20, or 30 this exceeds the number of tolerable errors and the method fails. The simulation shows that the clustering of errors in bursts is, in fact, beneficial when the BER remains

Table 2: Message parameters for the SAE benchmark example.

Msg	DLC (b)	T_i (ms)	D_i (ms)	J_i (ms)	C_i (ms)
1	1	50.000	5.000	0.100	0.273
2	2	5.000	5.000	0.100	0.303
3	1	5.000	5.000	0.100	0.273
4	2	5.000	5.000	0.100	0.303
5	1	5.000	5.000	0.100	0.273
6	2	5.000	5.000	0.100	0.303
7	6	10.000	10.000	0.200	0.424
8	1	10.000	10.000	0.200	0.273
9	2	10.000	10.000	0.200	0.303
10	3	10.000	10.000	0.200	0.333
11	1	50.000	20.000	0.200	0.273
12	1	100.000	100.000	0.300	0.273
13	4	100.000	100.000	0.300	0.364
14	1	100.000	100.000	0.200	0.273
15	3	1000.000	1000.000	0.400	0.333
16	1	1000.000	1000.000	0.300	0.273
17	1	1000.000	1000.000	0.300	0.273

constant as errors tend to cluster together and their effects are less pronounced. This behavior is clearly captured by the methods proposed in this paper.

8. Application to SAE Benchmark Message Set

In this section, an extended example based upon a benchmark set of CAN messages is provided to further illustrate the features of the proposed technique. The well-known "SAE benchmark set" of $n = 17$ messages under consideration is as described in [32] and consists of over 50 typical automotive signals packed into 17 periodic/sporadic messages. In this paper, we assume that messages are fixed priority (DMJM) scheduled using CAN in its native form. Assuming a bit rate of 330 Kbps and extended 29-bit (cf. standard 11 bits) message identifiers, the messages and their parameters are as shown in Table 2. The total utilization of this CAN network is 44.5% in the absence of errors. The messages have been ordered according to nondecreasing deadline minus jitter.

To illustrate the full use of the methods proposed in this paper, some realistic assumptions were first made upon the BER and burst distribution. The BER was selected to have a value 1×10^{-6}, with two distributions of bursts considered: a static configuration with errors arriving independently and a burst configuration with errors arriving in bursts of mean length 5. This configuration is not dissimilar with typical error rates and burst characteristics reported in the literature for CAN networks and employed in previous studies [12, 25]. An error frame of $C_E = 31$ bits was employed. Bounds on p_i^{fail} were then determined for fixed priority (DMJM) scheduling, applying the techniques developed in Sections 4, 5, and 6. The values obtained are shown in Table 3. Also indicated are the results obtained from application of the method described in [8]. Entries in the table given as "$\approx 0.00E + 00$" indicate that numerically the event occurs with such a low probability that it could not be represented with double-precision floating point accuracy.

From this table, some interesting observations can be made. In all cases, the new bound described in this paper gave deadline failure bounds smaller than the previous method described in [8]. In the case of static errors, both bounds predict extremely low deadline failure probability for all messages, that is, an indication that such events should be extremely rare. The reduced failure rate in the new method is due to the relaxation of the assumption that all errors strike the last bit of the longest frame transmitted in the considered interval.

Considering next the burst case, one observes drastic differences between the two methods. In particular, the previous work [8] predicts that, as the burst length increases for a given BER, the probability of deadline misses increases significantly; in fact, the method fails to produce results for messages 5, 6, and 10. In these cases, the critical number of errors required to cause a deadline miss exceeds the mean number of errors arriving in the $(D - J)$ interval. This is not as predicted by the new method, which predicts that the failure probability should be decreasing for fixed BER; this is again due to the relaxation of the assumption that all errors strike the last bit of the longest frame transmitted in the considered interval and the classification of errors into those of type 1 and type 2. Note that these assumptions have been validated using stochastic simulations in the previous section. In particular, consider the highest priority message (Msg 1), which has length 90 bits and a deadline slack of 1,387 bits. Under error bursts, the deadline failure probability is bounded to be 0.0118 using the previous method: a comparatively commonplace event occurring once every 4.24 seconds for the given message period. However in the previous section, recall that under simulation a longer (135 bits) CAN message with shorter deadline slack (365 bits) was found to miss its deadline with probability 0.000008529 in a far more aggressive environment with longer burst durations. As such, the quantification by the new method where a deadline miss remains to be an extremely rare event even in a burst environment seems much more reasonable, although further detailed simulations are required to confirm this. Finally, note that the calculations necessary to obtain these results were embedded in a simple spreadsheet, giving a good indication of the ease of application of the method and that it is a good candidate for use in the stated area of motivation in Introduction.

9. Conclusions

In this paper, a simple and efficient method to bound the deadline failure probabilities for a real-time industrial CAN network transmitting periodic/sporadic messages which may experience release jitters under fixed or dynamic priority scheduling has been proposed. As demonstrated through stochastic simulation, indications are such that the procedure is effective at capturing the effects of errors and burst distributions on the message schedule and produces results which are more accurate and useful than previous methods. A representative example consisting of a benchmark

Table 3: Deadline failure probabilities for static and bursty environments.

Msg	p^{fail} (old bound [8])		p^{fail} (new bound)	
	Static	Bursts	Static	Bursts
1	$1.21E-39$	$1.18E-02$	$8.31E-44$	$8.13E-52$
2	$5.24E-32$	$6.80E-02$	$8.33E-37$	$1.27E-43$
3	$2.96E-28$	$1.51E-01$	$5.51E-31$	$8.24E-37$
4	$6.54E-21$	$4.55E-01$	$3.07E-25$	$4.56E-30$
5	$2.47E-17$	>0.5	$1.18E-19$	$1.73E-23$
6	$7.83E-14$	>0.5	$3.37E-14$	$4.89E-17$
7	$8.02E-26$	$1.51E-01$	$4.02E-29$	$5.03E-35$
8	$2.01E-22$	$3.32E-01$	$3.95E-24$	$4.19E-29$
9	$4.42E-19$	$4.55E-01$	$4.70E-20$	$2.95E-24$
10	$1.32E-12$	>0.5	$1.19E-15$	$5.33E-19$
11	$7.99E-43$	$5.43E-04$	$4.88E-49$	$2.86E-59$
12	$1.25E-307$	$4.59E-91$	$\approx 0.00E+00$	$\approx 0.00E+00$
13	$3.82E-304$	$9.16E-90$	$\approx 0.00E+00$	$\approx 0.00E+00$
14	$1.27E-300$	$1.81E-88$	$\approx 0.00E+00$	$\approx 0.00E+00$
15	$\approx 0.00E+00$	$\approx 0.00E+00$	$\approx 0.00E+00$	$\approx 0.00E+00$
16	$\approx 0.00E+00$	$\approx 0.00E+00$	$\approx 0.00E+00$	$\approx 0.00E+00$
17	$\approx 0.00E+00$	$\approx 0.00E+00$	$\approx 0.00E+00$	$\approx 0.00E+00$

message set for a CAN network has helped to illustrate key elements of the proposed technique and highlight its advantages over a recent similar method. In future work, more detailed simulation studies will be carried out to explore the accuracy of the proposed bound. In addition, extensions of this bounding technique will be used, in conjunction with appropriate environmental error monitoring techniques, to move towards efficient dependability-aware QoS controls for flexible real-time CAN networks.

Appendix

A.

Consider the case where a specified proportion of type 1 errors $p_c \in [0, 1]$ will manifest only as CRC failures signaled by dropping the current transmission and commencing an error frame after the CRC check. Let the number of lost bits following a CRC failure be given by C_C. Incorporating this information into (3) gives a modified expression for the mean load introduced at each time-step:

$$E[e(k)] = p_g \cdot \left(\left((1-p_c)\cdot\frac{C+1}{2} + p_c \cdot C_C\right) + C_E\right) + p_b. \tag{A.1}$$

An expression for the variance of $e(k)$ can once again be written using the "mean of square minus square of the mean" rule and proceeding as before:

$$\begin{aligned}\mathrm{Var}(e(k)) &= \left(\frac{(1-p_c)\cdot p_g}{C}\cdot\left(\sum_{i=1}^{C}(i+C_E)^2\right)\right) \\ &\quad + \left(p_g \cdot p_c \cdot (C_C + C_E)^2\right) + \left(p_b \cdot 1^2\right) \\ &\quad - E[e(k)]^2 \\ &= p_g \cdot (1-p_c) \\ &\quad \cdot \left(\frac{C^2}{3} + \frac{C}{2} + \frac{1}{6} + C_E^2 + C_E \cdot (C+1)\right) \\ &\quad + p_g \cdot p_c \cdot (C_C + C_E)^2 + p_b \\ &\quad - E[e(k)]^2.\end{aligned} \tag{A.2}$$

The above expressions can also be used in situations in which every type 1 error is taken to lead to the same (constant) error load (as in previous analysis for CAN, this could be the worst-case length of the longest message). This is achieved by setting $p_c = 1$ and C_C to the required number of bits.

B.

Proof of (14). The expression is proved by contradiction, using a similar approach to [31]: if the messages are not schedulable under EDF, it will be shown that the condition of (14) is false. Assume that a deadline miss occurs in the schedule and the condition of (14) holds. Let the first deadline miss in the schedule occur at time t_2, and let the first time the network was idle (or occupied processing a frame with deadline $> t_2$) be time t_1. Without loss of generality, assume that $t_1 = 0$ and set $t = (t_2 - t_1)$. As it was assumed that the messages are sorted in order of nondecreasing deadline minus jitter, for convenience let the highest-indexed message satisfying the relationship $(D_k - J_k) \le t$ be given by i. Observe that, under EDF scheduling, no message with index $> i$ can contribute to the network demand in the interval $[0, t)$ and that there is no idle time in this interval [7]. Assuming that each message

with index $j \leq i$ is activated at time $t = -J_j$ to maximize the network loading, then the network demand $h(t)$ can then be calculated using [7]:

$$h(t) = \sum_{j=1}^{i} \left(\left\lfloor \frac{t - D_j + J_j}{T_j} \right\rfloor + 1 \right) \cdot C_j + \max_{(D_k - J_k) > t} \{C_k\}. \tag{B.1}$$

Since the deadline miss occurs at time t, the demand in the interval $[0, t)$ must exceed the length of the interval and we can write

$$\begin{aligned} t &< \sum_{j=1}^{i} \left(\left\lfloor \frac{t - D_j + J_j}{T_j} \right\rfloor + 1 \right) \cdot C_j + \max_{(D_k - J_k) > t} \{C_k\} \\ &\leq \sum_{j=1}^{i} \left(\frac{t - D_j + J_j}{T_j} \cdot C_j + C_j \right) + \max_{(D_k - J_k) > t} \{C_k\} \\ &= \sum_{j=1}^{i} U_j \left(t - D_j + J_j + T_j \right) + \max_{(D_k - J_k) > t} \{C_k\} \\ t \left(1 - \sum_{j=1}^{i} U_j \right) &< \sum_{j=1}^{i} U_j \left(D_j + J_j + T_j \right) + \max_{(D_k - J_k) > t} \{C_k\}. \end{aligned} \tag{B.2}$$

Since $(D_i - J_i) \leq t$ we can write

$$\begin{aligned} (D_i - J_i) \left(1 - \sum_{j=1}^{i} U_j \right) &< \sum_{j=1}^{i} U_j \left(D_j + J_j + T_j \right) + \max_{(D_k - J_k) > t} \{C_k\} \\ (D_i - J_i) &< \sum_{j=1}^{i} U_j \left(D_i - J_i - D_j + J_j + T_j \right) + B_i, \end{aligned} \tag{B.3}$$

where B_i is as defined in (15). Collecting terms in a more convenient manner for computation gives rise to (14):

$$D_i < J_i \left(1 - \sum_{j=1}^{i} U_j \right) + \sum_{j=1}^{i} U_j J_j + B_i + D_i \sum_{j=1}^{i} U_j + \sum_{j=1}^{i} U_j \left(T_j - D_j \right). \tag{B.4}$$

Thus by checking that for each message relative deadline (14) holds, a deadline miss will not occur as claimed. □

Competing Interests

The author declares that there are no competing interests regarding the publication of this paper.

Acknowledgments

Preliminary versions of some of the ideas presented in this paper were presented at the 19th IEEE International Conference on Emerging Technology & Factory Automation [8].

References

[1] M. Short and J. Proenza, "Towards efficient probabilistic scheduling guarantees for real-time systems subject to random errors and random bursts of errors," in *Proceedings of the 25th Euromicro Conference on Real-Time Systems (ECRTS '13)*, pp. 259–268, Paris, France, July 2013.

[2] J. C. Laprie, *Dependability: Basic Concepts and Terminology*, Springer, New York, NY, USA, 1992.

[3] R. Bosch, *CAN Specification 2.0*, Robert Bosch GmbH, Stuttgart, Germany, 1991.

[4] P. Pedreiras and L. Almeida, "EDF message scheduling on controller area network," *Computing & Control Engineering Journal*, vol. 13, no. 4, pp. 163–170, 2002.

[5] M. D. Natale, "Scheduling the CAN bus with earliest deadline techniques," in *Proceedings of the 21st IEEE Real-Time Systems Symposium (RTSS '00)*, pp. 259–268, December 2000.

[6] A. Zuhily and A. Burns, "Optimality of (D-J)-monotonic priority assignment," *Information Processing Letters*, no. 103, pp. 247–250, 2007.

[7] J. A. Stankovic, M. Spuri, K. Ramamritham, and G. C. Buttazzo, *Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms*, Kluwer Academic Publishers, 1998.

[8] M. Short, "Simple bounds on deadline failure probabilities in fault-tolerant real-time networks," in *Proceedings of the 19th IEEE International Conference on Emerging Technology & Factory Automation*, Barcelona, Spain, September 2014.

[9] A. Burns, S. Punnekkat, L. Strigini, and D. Wright, "Probabilistic scheduling guarantees for fault-tolerant real-time systems," in *Proceedings of the Dependable Computing for Critical Applications*, pp. 361–378, IEEE, San Jose, Calif, USA, November 1999.

[10] H. Aydin, "Exact fault-sensitive feasibility analysis of real-time tasks," *IEEE Transactions on Computers*, vol. 56, no. 10, pp. 1372–1386, 2007.

[11] M. Jonsson and K. Kunert, "Towards reliable wireless industrial communication with real-time guarantees," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 4, pp. 429–442, 2009.

[12] N. Navet, Y.-Q. Song, and F. Simonot, "Worst-case deadline failure probability in real-time applications distributed over controller area network," *Journal of Systems Architecture*, vol. 46, no. 7, pp. 607–617, 2000.

[13] H. A. Hansson, T. Nolte, C. Norström, and S. Punnekkat, "Integrating reliability and timing analysis of CAN-based systems," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 6, pp. 1240–1250, 2002.

[14] I. Broster, A. Burns, and G. Rodríguez-Navas, "Comparing real-time communications under electromagnetic interference," in *Proceedings of the 16th Euromicro Conference on Real-Time Systems*, pp. 45–52, July 2004.

[15] F. Many and D. Doose, "Scheduling analysis under fault bursts," in *Proceedings of the 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '11)*, pp. 113–122, Chicago, Ill, April 2011.

[16] L. Marques, V. Vasconcelos, P. Pedreiras, and L. Almeida, "Comparing scheduling policies for a message transient error recovery server in a time-triggered setting," in *Proceedings of the 19th IEEE International Conference on Emerging Technology & Factory Automation*, Barcelona, Spain, September 2014.

[17] A. Gujarati and B. B. Brandenburg, "When is CAN the weakest link? A bound on failures-in-time in CAN-based real-time systems," in *Proceedings of the 36th IEEE Real-Time Systems Symposium (RTSS '15)*, San Antonio, Tex, USA, May 2015.

[18] G. I. Mary, Z. C. Alex, and L. Jenkins, "Response time analysis of messages in controller area network: a review," *Journal of Computer Networks and Communications*, vol. 2013, Article ID 148015, 11 pages, 2013.

[19] M. Short, I. Sheikh, and S. A. I. Rizvi, "Bandwidth-efficient burst error tolerance in TDMA-based CAN networks," in *Proceedings of the IEEE 16th Conference on Emerging Technologies and Factory Automation (ETFA '11)*, pp. 1–8, Paris, France, September 2011.

[20] P. Pedreiras, P. Gai, L. Almeida, and G. C. Buttazzo, "FTT-Ethernet: a flexible real-time communication protocol that supports dynamic QoS management on ethernet-based systems," *IEEE Transactions on Industrial Informatics*, vol. 1, no. 3, pp. 162–172, 2005.

[21] P. Bartolomeu, J. Fonseca, and F. Vasques, "Implementing the wireless FTT protocol: a feasibility analysis," in *Proceedings of the 15th IEEE International Conference on Emerging Technologies & Factory Automation*, pp. 1–10, IEEE, Bilbao, Spain, September 2010.

[22] M. Collotta, G. Pau, and G. Scatà, "Deadline-aware scheduling perspectives in industrial wireless networks: a comparison between IEEE 802.15.4 and bluetooth," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 602923, 11 pages, 2013.

[23] R. I. Davis and A. Burns, "Response time upper bounds for fixed priority real-time systems," in *Proceedings of the Real-Time Systems Symposium (RTSS '08)*, pp. 407–418, Barcelona, Spain, December 2008.

[24] E. Tran, "Multi-bit error vulnerabilities in the controller area network protocol," Research Report Series, Institute for Complex Engineered Systems, Carnegie Mellon University, Pittsburgh, Pa, USA, 1999.

[25] J. Ferreira, A. Oliveira, P. Fonseca, and J. A. Fonseca, "An experiment to assess bit error rate in CAN," in *Proceedings of the 3rd International Workshop on Real-Time Networks*, Catania, Italy, June 2004.

[26] E. N. Gilbert, "Capacity of a burst-noise channel," *The Bell System Technical Journal*, vol. 39, pp. 1253–1265, 1960.

[27] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *Proceedings of the18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99)*, vol. 1, pp. 345–352, New York, NY, USA, March 1999.

[28] G. Bennett, "Probability inequalities for the sum of independent random variables," *Journal of the American Statistical Association*, vol. 57, no. 297, pp. 33–45, 1962.

[29] F. Eisenbrand and T. Rothvoß, "EDF-schedulability of synchronous periodic task systems is coNP-Hard," in *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SoDA '10)*, pp. 1029–1035, Austin, Tex, USA, January 2010.

[30] F. Eisenbrand and T. Rothvoß, "Static-priority real-time scheduling: response time computation is NP-hard," in *Proceedings of the Real-Time Systems Symposium (RTSS '08)*, pp. 397–406, IEEE, Barcelona, Spain, December 2008.

[31] U. C. Devi, "An improved schedulability test for uniprocessor periodic task systems," in *Proceedings of the 15th Euromicro Conference on Real-Time Systems ((ECRTS '03)*, pp. 23–30, IEEE, July 2003.

[32] K. Tindell and A. Burns, "Guaranteeing message latencies on Controller Area Network (CAN)," in *Proceedings of the 1st International CAN Conference*, Mainz, Germany, 1994.